

## Refine Search

### Search Results -

Terms	Documents
705/35	2322

Database:

US Pre-Grant Publication Full-Text Database  
 US Patents Full-Text Database  
 US OCR Full-Text Database  
 EPO Abstracts Database  
 JPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

Search:






### Search History

 DATE: Tuesday, February 28, 2006    [Printable Copy](#)    [Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u> result set
side by side			
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>			
<u>L14</u>	705/35	2322	<u>L14</u>
<u>L13</u>	705/1	5512	<u>L13</u>
<u>L12</u>	707/1	7538	<u>L12</u>
<u>L11</u>	L10 and display	15	<u>L11</u>
<u>L10</u>	L9 and prices	16	<u>L10</u>
<u>L9</u>	L8 and merchant	37	<u>L9</u>
<u>L8</u>	L7 and (database or data with base)	50	<u>L8</u>
<u>L7</u>	L6 and credit near card near transaction	56	<u>L7</u>
<u>L6</u>	705/30	1011	<u>L6</u>
<u>L5</u>	L3 and (credit near card or credit with card or credit adj card or debit near card or debit with card or debit adj card)	6	<u>L5</u>
<u>L4</u>	L3 and transaction	14	<u>L4</u>
<u>L3</u>	L2 and (key with definition or key near definition or key adj definition)	19	<u>L3</u>

<u>L2</u>	L1 and key near elements	67	<u>L2</u>
<u>L1</u>	(object-oriented or objectoriented or object adj oriented) near (data with base or database)	3689	<u>L1</u>



END OF SEARCH HISTORY

## Freeform Search

---

<b>Database:</b>	US Pre-Grant Publication Full-Text Database
	<b>US Patents Full-Text Database</b>
	US OCR Full-Text Database
	EPO Abstracts Database
	JPO Abstracts Database
	Derwent World Patents Index
	IBM Technical Disclosure Bulletins

<b>Term:</b>	<input type="text"/>	 
--------------	----------------------	--

<b>Display:</b>	<input type="text" value="10"/>	<b>Documents in Display Format:</b>	<input type="text" value="TI"/>	<b>Starting with Number</b>	<input type="text" value="1"/>
-----------------	---------------------------------	-------------------------------------	---------------------------------	-----------------------------	--------------------------------

**Generate:** ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

---

Search

Clear

Interrupt

---

### Search History

---

**DATE:** Tuesday, February 28, 2006    [Printable Copy](#)    [Create Case](#)

<u>Set</u> <u>Name</u> <u>Query</u> side by side	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u> result set
<i>DB=USPT; PLUR=YES; OP=OR</i>		
<u>L6</u> (5493671   5388259   5058000   5794246   5687363   5522066   4961139   5608904   5724569   5513348   5680618   5428782   5802511   5734915   5701466   5263159   5708828   5446883   5210824   5495606   5509136   5628003)! [PN]	22	<u>L6</u>
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L5</u> ('6023694')[ABPN1,NRPN,PN,TBAN,WKU]	2	<u>L5</u>
<u>L4</u> 6023694.pn.	2	<u>L4</u>
<u>L3</u> 5950192.pn.	2	<u>L3</u>
<u>L2</u> 5806058.pn.	2	<u>L2</u>
<u>L1</u> 5553218.pn.	2	<u>L1</u>

END OF SEARCH HISTORY

## Freeform Search

**Database:** US Pre-Grant Publication Full-Text Database  
 US Patents Full-Text Database  
 US OCR Full-Text Database  
 EPO Abstracts Database  
 JPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

**Term:**

**Display:**  **Documents in Display Format:**  **Starting with Number**

**Generate:** ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

### Search History

**DATE:** Tuesday, February 28, 2006 [Printable Copy](#) [Create Case](#)

<u>Set Name</u> side by side	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u> result set
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L20</u>	L19 and pric\$	91	<u>L20</u>
<u>L19</u>	L18 and (merchant or vendor or supplier)	101	<u>L19</u>
<u>L18</u>	L17 and (credit with card or credit near card or credit adj card)	118	<u>L18</u>
<u>L17</u>	L16 and transaction	185	<u>L17</u>
<u>L16</u>	L15 and key with definition	185	<u>L16</u>
<u>L15</u>	L14 and key with elements	1273	<u>L15</u>
<u>L14</u>	(transaction with database or transaction near database or transaction adj database or transaction with data with base or transaction near data with base or transaction adj data with base)	22733	<u>L14</u>
<u>L13</u>	L12 and (definition or define or defin\$)	41	<u>L13</u>
<u>L12</u>	l9 and (key with elements or key near elements or key adj elements)	41	<u>L12</u>
<u>L11</u>	l7 and 707.clas.	46	<u>L11</u>
<u>L10</u>	l9 and 707.clas.	14	<u>L10</u>
<u>L9</u>	L7 and display near screen	188	<u>L9</u>
<u>L8</u>	L7 and 705.clas.	312	<u>L8</u>

<u>L7</u>	L6 and (pric\$ with values or pric\$ near values or pric\$ adj values)	564	<u>L7</u>
<u>L6</u>	L5 and (merchant or supplier or vendor)	3982	<u>L6</u>
<u>L5</u>	L4 and transaction	6896	<u>L5</u>
<u>L4</u>	L2 and (credit with card or credit near card or credit adj card)	11359	<u>L4</u>
<u>L3</u>	L2 and transaction	46384	<u>L3</u>
<u>L2</u>	(relational or relation) or (object-oriented or objectoriented or object adj oriented) near (data with base or database)	1982987	<u>L2</u>
<u>L1</u>	(relational or relation) near (data with base or database)	21738	<u>L1</u>

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set

☐ [Generate Collection](#) [Print](#)

L20: Entry 91 of 91

File: USPT

Jan 19, 1999

DOCUMENT-IDENTIFIER: US 5862325 A

TITLE: Computer-based communication system and method using metadata defining a control structure

Abstract Text (1):

An automated communications system operates to transfer data, metadata and methods from a provider computer to a consumer computer through a communications network. The transferred information controls the communications relationship, including responses by the consumer computer, updating of information, and processes for future communications. Information which changes in the provider computer is automatically updated in the consumer computer through the communications system in order to maintain continuity of the relationship. Transfer of metadata and methods permits intelligent processing of information by the consumer computer and combined control by the provider and consumer of the types and content of information subsequently transferred. Object oriented processing is used for storage and transfer of information. The use of metadata and methods further allows for automating many of the actions underlying the communications, including communication acknowledgements and archiving of information. Service objects and partner servers provide specialized data, metadata, and methods to providers and consumers to automate many common communications services and transactions useful to both providers and consumers. A combination of the provider and consumer programs and databases allows for additional functionality, including coordination of multiple users for a single database.

Drawing Description Text (40):

FIG. 38 is a block flow diagram for a process for executing payment transactions using service objects and partner servers.

Drawing Description Text (41):

FIG. 39 is a block flow diagram for a process for returning a transaction receipt using service objects and partner servers.

Detailed Description Text (3):

There is illustrated in FIG. 1 a first embodiment of a system of the present invention which automatically updates a database in a consumer computer with information from a provider computer. Numerous providers and consumers exist in the system of the present invention. However, since all communications can be separated into transfers between a single provider and consumer, the design and operation of the system is illustrated with only one provider and one consumer, except as otherwise noted. As illustrated in FIG. 1, a provider computer 1 includes a provider database 11 of communications control information which it desires to disseminate or make accessible to one or more consumers. A consumer computer 2 includes a consumer database 21 of communications control information received from providers and stored by the consumer. The organization, structure, and content of the provider database 11 and consumer database 21 are discussed below. The provider computer 1 is connected through a communications network 3 to the consumer computer 2. Any communications network 3 may be used to connect the provider computer 1 and the consumer computer 2, including direct network connections, server-based

environments, telephone networks, the Internet, intranets, local area networks (LANs), wide area networks (WANs), the World Wide Web, other webs, and even transfers of data on physical media such as disks or computer-readable paper outputs via postal communications networks. The particulars of the communications network illustrated as preferred embodiments are not limiting features of the invention. However, the Internet and World Wide Web provide existing capabilities between computers sufficient to provide the necessary connections. For this reason, the description of the present invention is based on this communications medium, which should be understood to be used for purpose of illustration only. Organization and operation of the Internet and communications over the Internet are discussed generally in Kris Jamsa and Ken Cope, Internet Programming (1995) and Marshall T. Rose, The Internet Message: Closing the Book with Electronic Mail (1993), which are incorporated herein by reference. Communications over the World Wide Web are discussed generally in John December and Neil Randall, The World Wide Web Unleashed (1996), which is incorporated herein by reference. Additionally, the illustrated embodiment is not limited to the specific networks known as the "Internet" and "World Wide Web", but relate to internet, intranet and web networks generally. A specific feature of this invention is that it is easily adaptable to control and automate communications via any type of communications network. In addition, it can select a preferred communications network and message encoding format to be used for a specific communications transaction, as further described below.

Detailed Description Text (57):

The event 116 class is an abstract class defining the attributes for scheduled events 117 and logged events 118. The scheduled event 117 class is used to create a queue of events for the provider program 12 or consumer program 22 to execute at some time in the future. An example is the polling operation necessary to update communications objects via the pull technique. The logged event 118 class is the counterpart used to create a log of past events. System events may need to be tracked for purposes of accumulated statistics, tracking user or communications object activity, documenting errors, providing payment transaction receipts, etc. Scheduled event and logged event objects can be further understood in the discussion of event loops, event logging, and event scheduling below.

Detailed Description Text (119):

The user can generate other reports relating to the consumer database using the other reports form 640. Standard reports might include database statistics (total objects, pages and elements; database file size; and size of objects being held), object statistics (frequency of use; last use; age in system; total age; size; number of updates; and last update), and transaction logs (number of updates; percentage of CPU time used, online time used; percentage of errors; and types of errors). Additionally, consumers could specify their own database reports to be added to this form.

Detailed Description Text (191):

As with other forms of encoding, communications objects are an excellent mechanism for simplifying and automating public/private key encryption. Referring to the data structures in FIG. 3, this is because a communications object 110 is an ideal vehicle for transmitting one or more of the provider's public keys to the consumer's computer, where it can be used to automatically encrypt messages being returned to the provider. The public key can be stored as an element 143, and the encryption method can be stored as a method 141. By encrypting the return message as a message object 110, the message object can invoke a receipt method 141 at the provider program 12 which can automatically decrypt the message using the provider's private key and the decryption method, stored as an element 143 and method 141 in the provider database 11, or otherwise made available to the receipt method 141.

Detailed Description Text (193):

One particular advantage of a communications object system in this respect is the ease with which multiple public keys may be used. Multiple keys may be included within a single communications object, or a single key may be constantly changed via communications object updates, or both techniques can be used together. Since encryption can be applied automatically by the consumer program 22, the encryption method 141 can programatically or randomly chose from among the available public keys. By including an identifier value 161 within each public key element 143, and including this unencrypted identifier value in the header of the encrypted message objects 110, the provider program 12 can also automatically identify and apply the matching private key element 143 for decryption. The use of multiple rotating public keys significantly reduces the risk of security breaches if any one key combination is broken, and increases the effort necessary to compromise the security of the messages.

Detailed Description Text (247):

The ability of a communications object system to automate common communications tasks is perhaps best exemplified by its ability to automate data exchanges between consumers and providers. Typical examples include the exchange of contact information, demographic data, psychographic data, billing information, product registration information, customer service data, technical support data, transaction histories, stock feeds, news data, weather data, and so on. A communications object system is capable of automating the exchange of such data to a greater degree than any other existing communications system for five reasons. First, such data is already stored in a consumer database 21 in such a fashion as to be available for automated access and delivery. Second, such data is available in structured, typed formats that allow providers to easily specify the data they require. Thirdly, communications objects give providers the tool they need to transfer such data from the consumer back to the provider. Fourth, message objects and the architecture of the provider program 12 allow the provider to automate the processing of such data when it is received back at the provider. Fifth, the ability of the provider program 12 and consumer program 22 to automatically trigger events and respond to message objects means a multi-part data exchange transaction (such as a purchase and receipt acknowledgment) can be automated throughout.

Detailed Description Text (250):

Data type control is required because providers need a way to specify the data they require in a specific data exchange transaction. The data type definition features of a communications object system, as explained above in the data structure section, are ideally suited to this need. By creating a system-wide set of low-level composite type definitions, such as Name, Address, and Telephone, and then nesting these inside of progressively more comprehensive composite type definitions, such as BusinessCard or Resume, a hierarchy of standard data type definitions can be created that are available to all providers and consumers. This has two very significant advantages. First, as providers design input forms and methods for data exchange tasks, they can choose from among these standard data type definitions rather than needing to create their own composite data type definitions, saving considerable time and effort. Second, data type standardization means that consumers need only enter data once into each instance of each data type that pertains to them. For example, the consumer only needs to enter his/her name, addresses, telephone numbers, birthdate, and other personal data one time into the consumer database 21. From that point on all communications objects which need data of these types can access these data type instances. This saves the consumer data input time and also vastly reduces the potential for data input errors.

Detailed Description Text (252):

Providers can also create their own data type definitions and specify the use of these composite data type definitions in data exchange methods. When a provider-specific data type can be aggregated or calculated from other system standard data type definitions which are already present in the consumer database 21, the resulting element can be composed automatically by a data exchange method. When a



provider-specific data type requires the input of new data from the consumer, an input form can be generated by the data exchange method. Once submitted, the data can also be saved as a element preference instance (147, FIG. 3) in the consumer database 21. The provider can then use the system ID of the type definition of this element to query for this element preference instance in future transactions. This allows a provider to dynamically generate and persistently store provider-specific data type definitions in the consumer database 21. A common example of such a data type might be a consumer's preference between a provider's selection of product colors, such as clothing or paint. Storing this data locally at the consumer database 21 means that it can easily be included in any future communications from the consumer. Additionally, such data can be shared among all communications objects or data exchange methods from that provider, as further explained below. Another key benefit is that this data can be easily and immediately edited by the customer should the customer's information or preferences change. Such changes can also be automatically transmitted back to the provider through the use of data association rules, discussed below.

Detailed Description Text (253):

As with any multiuser database system, shared access to data requires data access controls. This control should cover all common data operations such as creating, reading, writing, moving, and deleting data. In a communications object system, data access controls need to extend beyond human operators to communications objects, since these objects are essentially acting as "surrogates" for their respective providers. The key data structure involved with data access control is the rules class 140. Data access rules can monitor all forms of data access within the provider database 11 or consumer database 21 as well as external data in the provider or consumer's computing or network environment. For example, a typical rule governing access to communications object components or element preference instances might be that only other communications objects sharing the same database system ID (100, FIG. 3) can read, write, or delete such instances. This would prevent different providers from having access to each other's private data. This rule could be modified so that only communications objects sharing a group system ID (251, FIG. 6A), described above, could have access to such data. This would allow all communications objects created by employees of the same company, or within a company division, to access each other's communications object component or element preference instances. Data access rules can be system-wide, assigned by providers, or assigned by consumers. An example of a provider-assigned rule would be restrictions on communications object forwarding, which will be further discussed below. An example of a consumer-assigned rule would be that designated personal data, such as household income, must be explicitly authorized by the consumer before it is transmitted in any data exchange. A stricter rule would state that more sensitive private data, such as credit card numbers, must be encrypted and require one or more passkeys for decryption prior to any data exchange. In order to protect their integrity, data access rules can also enforce the ability to add or change other data access rules, and also the hierarchy in which rules take precedence when two or more rules apply. Data access rules can also be selectively applied by the consumer to particular communications objects 110 or communications object groups such as folders 115 by creating associations between these and a data access rule 140. The application of rules to control data access within an active database is further discussed in the aforementioned Active Database Systems.

Detailed Description Text (259):

One of its most powerful forms of data exchange control in a communications object system is the ability to automate external data queries and the processing of query result sets. This is because it gives providers a tool to allow consumers quickly and easily set up automated queries against any type of data server maintained by the provider. These queries are easily set up because they can be composed using any data available in the consumer database 21 (subject to the consumer's data access rules, as explained above), so the consumer need only enter any new data required. The queries are easily automated because the data exchange method that

executes them can create its own scheduled event instances (117, FIG. 3) to execute future instances of the query. External query control can also be combined with notification control to automate notification depending on the query results. For example, a data exchange method that executes a data query for a stock price can notify the consumer if the new price is a certain dollar amount or percentage amount changed from the previous price.

Detailed Description Text (262):

By being able to control the exchange of external system data, file data, and data available via external queries in addition to internal data, the programs 12, 22 can automate many routine information transactions on data communications networks. This can produce a vast savings in the human labor normally required to exchange such data. The present invention is able to further increase this labor savings by automating the processing of such data once it has been exchanged. As with other data exchange operations, this is accomplished through the use of data exchange elements 143, data exchange methods 141, and message objects 110. Any data exchange method can produce a message object 110 that can call itself or another method or methods for processing the contents of the message object once it is received. As explained above, data exchange methods that call themselves are polymorphic, performing different operations at the provider program 12 than at the consumer program 22. An example of such a method is the SendAck method discussed above. Like any communications object method, data exchange methods can also call other methods, including other data exchange methods. In this way a succession of automated data exchanges may take place between a provider program 12 and consumer program 22 without any human intervention if none is required. Such automated data exchanges may also occur between the provider program 12 or consumer program 22 and other data servers as explained in the discussion of data query control above and the sections on service object partner servers below. This includes requesting data from the server or posting data to the server.

Detailed Description Text (301):

As with forwarding control, transfer control can also be exerted by the original provider using transfer rules 140 and transfer methods 141 in the communications object 110. Transfer rules and transfer methods are a particularly powerful means of data exchange control because they can accomplish automatic data exchange events involving the provider, the transferring consumer, and the receiving consumer all in one. An example is the transfer of ownership of a real world object, such as an automobile. A real-world rule applies to such a transfer, namely that the selling consumer must notify the automobile licensing authority, and the buying consumer must apply for a new title from the licensing authority. In this case the licensing authority would be the provider of a communications object 110 representing an automobile title. The selling consumer would have obtained the communications object 110 when he/she purchased the automobile. Using data exchange methods as explained above, the licensing authority would have used the communications object 110 to obtain the necessary elements 143 from the consumer required by law to register the vehicle. The licensing authority could then use updates to the communications object 110 to communicate with the consumer about the license, such as sending notifications about annual license renewals. (Payment for such license renewals can also be automated by data exchange methods in the communications object 110, as further described below.) When the time came to transfer the automobile title, the selling consumer would invoke the transfer method 141 in the communications object 110. The transfer method 141 would first generate an input form requesting the necessary data about the buying consumer and transaction details. (If a communications object 110 representing the buying consumer was also present, an association with such object 110 could be used to provide such data.) The transfer method 141 would then produce two message objects 110. The first message object would be transmitted to the licensing authority, containing the necessary elements 143 to automatically register the sale. The second message object would be transmitted to the buying consumer. This would include the forwarded communications object 110 representing the title. A transfer rule 140

would also determine which element preference instances 147 must be transferred with the communications object 110. For example, the Vehicle Identification Number (VIN) must be transferred with the title; a new VIN may not be specified by the buyer. The transfer method 141 would also add a rule 140 to the selling consumer's database 21 requiring that affirmative acknowledgment message objects needed to be received from both the licensing authority and the buying consumer before the communications object 110 representing the title will be deleted. The transfer method 141 could also create a scheduled event 117 that checked for the receipt of these message objects after a specified interval.

Detailed Description Text (303):

Transfer control can be applied to almost any situation where the real world ownership of an object or goods is transferred by an exchange of data between the transferee and the transferor, or between the transferee, transferor, and a third party such as a licensing authority, broker, agency, listing service, and so on. A universal example is classified ads. By using a communications object instance 110 to represent goods for sale via a classified advertising service, all or most of the communications transactions between the buyer, seller, and classified ad service can be automated. The use of a communications object system for classified advertising is further discussed in the description of data exchange service objects below.

Detailed Description Text (334):

Any given service object (815, FIG. 17) may provide services to providers, consumers, or both. Service objects that offer services to both providers and consumers are called polymorphic. Polymorphic service objects are particularly useful in a communications object system because many of the same services are required by both partners to a communications relationship, each in a different form depending on whether the partner is the provider or consumer. Such services typically fall into three categories: editing or searching databases, encoding or decoding communications, and automating transactions with third parties. An example of the first category is a directory service object, which permits providers to place or update listings in a directory service and permits consumers to automate searches of the same directory service for a specific provider. An example of the second category is an authentication service object, which permits providers to digitally sign communications objects and permits consumers to automatically verify these digital signatures. An example of the third category is a payment service object, which permits a provider to automate receiving payments from a bank or credit company and permits consumers to automate sending payments to the bank or credit company. Alternatively, where it is more efficient, service objects can be split into provider/consumer "pairs", each containing a link component object 110 linking it to the other.

Detailed Description Text (339):

As a subclass of standard communications objects 110, service objects can include all the control functions of communications objects described above. Certain control functions have special relevance for service objects. First, link control allows other communications objects to call the methods of a service object object regardless of where the service object may be located on a communications network 3. The special applications of link control will be discussed below. Second, update control allows a service object to stay current regardless of where it is located on a communications network 3. Version monitoring and update querying are particularly efficient techniques of update control for service objects and will be discussed below. Third, notification control allows a service object provider to notify providers or consumers using the service object about relevant changes to the service object or the communications services it makes available. Fourth, data exchange control allows the service object to automate data exchanges with the server or servers the service object may represent. Fifth, data archive control allows service objects to delete themselves if they age beyond a certain date or have not been used within a certain period. This allows databases 100 to avoid an

accumulation of seldom-used service objects. Finally, event tracking control and reporting control allows service objects to create and report transaction records which can be processed to provide further services to the provider or consumer. These transaction records can also be used by the service object provider for billing or statistical purposes.

Detailed Description Text (340):

Link control and update control have special applications to polymorphic service objects. The application of link control to polymorphic service objects is illustrated in FIG. 28. A provider using the provider program 12 has need of the services offered by a service object partner server 1302. First the provider obtains a service object 1310 from the partner server 1302 (step 1320). This may be by browsing with a web browser 50, receiving the service object 1310 via e-mail, or any of the other techniques described above. Such partner server 1302 may be a distribution server 32 or any other type of service object partner server such as those described below. Once the provider has obtained the service object 1310, the provider may add a link component object 110 to any of the provider's communications objects 110 which need to access the elements or methods the service object 1310. This link component object 110 will then be included in any communications object instance 35 generated from the consumer database 21 (step 1321). In a preferred embodiment, the link component object 110 is supplied by the service object 1310 itself. In this case, referring to FIG. 3, the provider need only create a contained-by association between the link component object 110 in the service object (1310, FIG. 28) and the provider's communications object 110. This association can also be created automatically when any service object method 141 is executed that creates a service relationship between the service object and the communications object 110. The communications object 110 thus becomes a synthesized object (813, FIG. 17), wherein the link component object 110 is supplied by the service object 1310. Examples of such a service object relationship include listing a communications object 110 in a directory server, registering a communications object 110 with an authentication server, or authorizing a communications object 110 for use with a payment server. Further examples will be given below. Referring again to FIG. 28, the next step is that a communications object instance 35 is transferred to a consumer program 22 (step 1322). This may be via e-mail using the push technique, via a distribution server 32 using the pull technique, or any of the other techniques described above. Once the communications object instance 35 is transferred to the consumer program 22, a link method 141 of the link component object 110 may be manually executed by the consumer or automatically executed by another system method or communications object method. For example, the consumer may wish to look up related communications objects instances 35 in a directory server, or authenticate the communications object instance 35 before forwarding it, or make a payment transaction using the communications object instance 35. When the link method 141 is executed, it uses the attributes of the link element 143 to locate the designated service object 1310 as described in the communications object exchange control section above. For example, if the service object 1310 is not present locally in the consumer database 21, the link method uses other attributes of the link element to locate the service object 1310. For instance, if a URL was present, the link method would use it to obtain the service object 1310. If this fails, the link method would use the UID or name of the service object 1310 to obtain its URL or other current network address via a name server. The link method could also call the methods of a name service object, described below. Once the link method located the proper network address, it would download the service object 1310 from the partner server 1302 (step 1323). At this point the link is reestablished, and the communications object instance 35 can call the service methods of the service object 1310 to perform the services requested (step 1324).

Detailed Description Text (384):

Many cryptographic protocols have been devised to provide authentication of user identity and message integrity over data networks. These include Kerberos 5, developed at MIT; SPX, developed by Digital Equipment Corporation; Privacy Enhanced

Mail (PEM), adopted by the Internet Engineering Task Force (IETF); Pretty Good Privacy (PGP), developed by Philip Zimmermann; and the CCITT X.509 protocols. Such protocols are fully described in the aforementioned Applied Cryptography by Bruce Schneier. Authentication service objects 1310 and authentication partner servers 1302 can be employed to automate the operation of many of these protocols. This is accomplished by storing the appropriate encryption keys as elements 143 and the appropriate encryption functions as methods 141 of the authentication service object 1310 or authentication partner server 1302.

Detailed Description Text (385):

An example is authentication using digital signatures based on public/private keys. The first set of steps in this process are shown in FIG. 32A. The process begins with the provider obtaining a suitable authentication service object (1310, FIG. 28) if one is not already present in the provider program 12 (step 4101). An authentication service object 1310 contains one or more public keys from its corresponding authentication partner server 1302, stored as elements 143. The authentication service object 1310 also contains the encoding method or methods 141 necessary to carry out its authentication functions, called authentication methods. When the provider is ready to create an authentication account, the provider executes one of the authentication methods 141 to generate a public/private key pair (step 4102). The private key is stored as an element 143 of the authentication service object 1310 in the provider database 11 (step 4103). Optionally, the data exchange method 141 may also encrypt this private key element 143 with a password known only to the provider and not stored anywhere in the provider database 11 or on the local computer. The authentication method 141 next creates an authentication order consisting of three elements: the public key generated in step 4102, the provider's UID, and a unique registration key known only to the provider and the authentication partner server 1302 (step 4104). Other elements or variables, such as a timestamp, may also be included. If the authentication partner server 1302 is operated in conjunction with a registration partner server 1302, the unique registration key may be the provider's password or other identification key created at the time of registration. This is shown as the Key attribute of the system ID instance (251, FIG. 6A). This unique registration key is stored in the provider database 11 as an encrypted element 143 which can be decrypted using a provider-supplied password. Alternatively, it may not be stored at all locally but be entered manually by the provider when required. The authentication method 141 next encrypts the authentication order using the authentication partner server's public key (step 4105). The authentication method 141 then creates a message object 110 containing the encrypted authentication order (step 4106). The authentication method 141 transmits this message object 110 to the authentication partner server 1302 (step 4107). The authentication partner server 1302 receives the message object 110 and executes its receipt method 141, which is either the same authentication method or another authentication method residing on the authentication partner server 1302 (step 4108). This authentication method 141 decrypts the authentication order using the authentication partner server's private key (step 4109). Next the authentication method 141 verifies the provider's unique registration key and UID in the authentication partner server database 1301 to validate the authentication order (step 4110). The authentication method 141 then creates a public key certificate by combining the provider's public key with certain other identifying data, such as the provider's UID (step 4111). The authentication method 141 digitally signs the public key certificate using the authentication partner server's private key (step 4112). The authentication method 141 then creates a message object 110 containing the public key certificate (step 4113). Finally, the authentication method 141 transmits the message object 110 back to the authentication service object 1310 in the provider program 12 (step 4114). There the provider program 12 receives the message object 110 and executes the original authentication method 141 in the authentication service object 1310 (step 4115). This authentication method 141 first verifies the signature of the public key certificate using the public key of the authentication partner server 1302 (step 4116). Lastly the authentication method 141 saves the public key certificate in the

provider database 11 as an element 143 (step 4117).

Detailed Description Text (387):

The final portion of the authentication process takes place when a communications object instance 35 bearing a digital signature arrives at a consumer program 22. These steps occur as part of the communications object receipt process, specifically as part of steps 721 or 722, FIG. 15. The steps in this process are illustrated in FIG. 32C. The process is initiated when a receipt method 141 of the communications object instance 35 calls an authentication method 141 in an authentication service object 1310 to verify the digital signature. First, the authentication method 141 uses the authentication partner server's public key, stored as an element 143 in the authentication service object 1310, to verify the digital signature on the provider's public key certificate (step 4131). Since the authentication partner server's private key was used to sign the certificate, only the authentication partner server's public key can be used to verify it. Once the public key certificate is authenticated, the authentication method 141 generates a hash of the communications object markup file using the same one-way hash algorithm used at the provider program 12 (step 4132). Finally, the authentication method uses the provider's public key to verify the provider's digital signature of the hash (step 4133). If the results of step 4132 and 1633 match, the communications object markup file is authenticated, and processing proceeds.

Detailed Description Text (389):

Authentication on a communications object system may also take place without using centralized authentication partner servers 1302. This technique, known as distributed key management, is used by the public-domain encryption program Pretty Good Privacy (PGP). It is based on the concept of an "introducer". An introducer is a person who signs the public key certificate of another person whose identity they personally know and are willing to certify. Introducers are easily employed on a communications object system using authentication service objects 1310. The steps in the process for using introducers are illustrated in FIG. 33A. First, a user requiring a public key certificate introduction, called the "originator", executes a data exchange method 141 of an authentication service object 1310 to generate a public/private key pair (step 4151). Next, the data exchange method 141 stores each key as an element 143 of the authentication service object 1310 (step 4152). Then the data exchange method 141 creates a public key certificate consisting of the public key element 143 plus such additional elements 143 as will allow any potential introducer to certify the identity of the originator (step 4153). These first three steps can be omitted if the originator only wishes to add introducers for an existing public key certificate already stored as an element 143 of the authentication service object 1310. Now, the data exchange method 141 generates an input form prompting the originator for the recipients 120 whom the originator would like to make introduction requests (step 4154). The checkboxes on this input form can represent each of the recipients 120 in the originator's consumer database 21, or the originator can specify the e-mail addresses of still other potential introducers. The input form also allows the originator to enter the attributes of a message element (211, FIG. 4) to be sent to these recipients. When the input form is submitted, the data exchange method 141 creates a message object 110 consisting of the public key certificate, the message element, and any other relevant data, such as a timestamp (step 4155). The data exchange method 141 transmits this to all recipients 120 selected by the originator (step 4156). When received by the recipient's consumer program 22, the message object's receipt method 141 executes the recipient's selected notification method or methods 141 for introduction requests (step 4157). If distributed key management was implemented on a communications object system, message objects containing introduction requests can use a standard notification element type definition 144. This type definition 144 allows consumers to assign notification methods 141 globally for all introduction requests, or designate specific notification methods for introduction requests from individual recipients 120. When the recipient responds to the notification message, a data exchange method 141 in the authentication service object 1310 is executed

(step 4158). This data exchange method 141 generates an input form for confirming the introduction request from the originator (step 4159). This input form may include any such data as may be relevant to an introduction request, including the elements 143 of the public key certificate that fully identify the originator. The recipient may also wish to verify the public key with the originator via another secure channel, such as via telephone. When the recipient is satisfied that the request is genuine, the recipient submits the input form (step 4160). The data exchange method 141 calls an authentication method 141 in the authentication service object 1310 which digitally signs the originator's public key certificate using the recipient's private key (step 4161). If the recipient's private key is stored as an encrypted element 143 of the authentication service object 1310, the recipient may need to enter password or passphrase for decryption. Then the data exchange method 141 creates a message object 110 containing the signed public key certificate (step 4162). The data exchange method 141 transmits this message object 110 to the originating authentication service object 1310 at the originating consumer program 22 (step 4163). When the message object 110 is received, the consumer program 22 executes the originating data exchange method 141 (step 4164). This data exchange method 141 stores the signed public key certificate as an element 143 of the authentication service object 1310 (step 4165). Finally, the data exchange method 141 executes any notification methods 141 assigned by the originator to the acknowledgment of introduction requests (step 4166).

Detailed Description Text (390):

Once a set of signed public key certificates has been received by the originator, the originator can send a public key acceptance request to any other communications object system user. The steps in the process for public key certificate acceptance requests are illustrated in FIG. 33B. The originator initiates the request by executing a data exchange method 141 of an authentication service object 1310 (step 4181). This data exchange method 141 generates an input form for the acceptance request (step 4182). This input form can include the attributes of a message element (211, FIG. 4) allowing the originator to compose the electronic equivalent of an introductory letter. The input form can also allow the originator to choose the introducers whose public key certificate signatures the originator wishes to present to the recipient. When the input form is submitted, the data exchange method 141 creates a message object 110 consisting of the selected public key certificate signatures, the message element, and any other relevant data, such as a timestamp (step 4183). Note that the first two steps above may be omitted if the acceptance request comes directly from another communications object method 141. In this case the recipient of the acceptance request will be specified in the method call, the set of introducer signatures can be selected algorithmically, and the message object in step 4183 can be created automatically. Next the message object 110 is transmitted to the recipient 120 (step 4184). When received by the recipient's consumer program 22, the message object's receipt method 141 executes a data exchange method 141 of an authentication service object 1310 (step 4185). This data exchange method 141 compares the UID of the introducer public key certificate signatures in the message object 110 with the UID of the trusted public key certificates stored in the recipient's consumer database 21 (step 4186). These trusted public key certificates are stored as elements 143 of the authentication service object 1310, and represent introducers whom the recipient trusts. For any matching UIDs, the data exchange method 141 then calls an authentication method 141 to verify the introducer signature using the introducer's public key (step 4187). The data exchange method 141 then checks an acceptance request preference element 147 in the recipient's consumer database 21 to determine if notification is desired (step 4188). For example, notification may not be desired if the signatures of 3 or more introducers are verified. If notification is desired, the data exchange method 141 executes the assigned notification methods 141 to generate a notification message for the recipient (step 4189). When the recipient responds to the notification message, a data exchange method 141 in the authentication service object 1310 is executed (step 4190). This data exchange method 141 generates an input form for confirming the acceptance request from the originator (step 4191).

This input form can include the results of the comparison test in step 4186. It can also include input fields for a message back to the originator, messages to the introducers, or other automated options. For purposes of this illustration, we will assume the recipient confirms the acceptance request when the input form is submitted (step 4192). (If the recipient denies the request, the following steps could produce a negative acknowledgment message to the originator.) The data exchange method 141 then saves the originator's public key certificate as an element 143 of the authentication service object 1310 (step 4193). This now becomes another of the recipients trusted public key certificates. The data exchange method 141 next creates a message object 110 containing an acknowledgment of the acceptance request (step 4194). Optionally, this message object 110 could also include a copy of the originator's public key certificate signed by the recipient using the recipient's private key. The data exchange method 141 transmits this message object 110 to the originating authentication service object 1310 at the originating consumer program 22 (step 4195). When the message object 110 is received, the consumer program 22 executes the originating data exchange method 141 (step 4196). This data exchange method 141 stores the public key certificate acceptance acknowledgment as an element 143 of the authentication service object 1310 (step 4197). Such acceptance acknowledgments can now be checked automatically by data exchange methods 141 in the consumer program 22. Alternatively, if the acceptance acknowledgment included a copy of the originator's public key certificate signed by the recipient, this public key certificate could be added to the originator's set of introducers. Finally the data exchange method 141 executes any notification methods 141 assigned by the originator to the acknowledgment of acceptance requests (step 4198).

Detailed Description Text (391):

These public key certificate introduction and acceptance processes can be further improved by the use of "trust levels". A trust level is one or more attributes of a public key certificate that indicate the level of trust the introducer extends to the originator. For example, a trust level attribute could accept an integer value range from 1 to 10, where 1 equals the lowest level of trust and 10 the highest level. The trust level is part of the public key certificate and is signed by the introducer so it cannot be modified by the originator. The trust level value can be entered by the introducer in step 4159 of FIG. 33A. Trust level values play the same role for public key certificates as threshold values play for notification elements, as explained in the notification control section. This means trust levels permit recipients to further automate the processing of acceptance requests and other operations pertaining to secure communications. This processing would take place in step 4186 of FIG. 33B. By implementing a trust rule 140, the recipient can determine what trust characteristics would qualify to generate an acceptance request automatically by the authentication service object 1310 without prior notification to the recipient. For example, a trust rule 140 could dictate that if an acceptance request had two or more verified introducers with trust levels of 8 or greater, an positive acknowledgment would be generated automatically. Trust rules 140 could also govern the autoexchange of signed public key certificates. For example, a trust rule could dictate that if an acceptance request had three or more verified introducers with trust levels of 9 or greater, the authentication service object 1310 would automatically sign and return a copy of the originator's public key certificate. Trust levels are a powerful technique for enabling efficient and effective distributed key management. Trust levels can also be used with other communications object system services such as feedback services, as described further below.

Detailed Description Text (393):

A final way authentication service objects 1310 can help ensure the security of a communications object system is security rules 140. Security rules can monitor all aspects of key handling and signature verification. Security rules can be particularly useful for enforcing a provider's control over forwarding or chaining of the provider's communications objects 110. When digital signatures do not match,



these rules can automatically trigger notification of the user of the programs 12, 22 via any notification method 141 the user desires. These rules can also generate message objects 110 capable of notifying the communications object provider, the authentication service object provider, and the communications object system vendor. Since authentication service objects play such a central role in the security of a communications object system, they can be subject to special rules 140. For example, a rule may require one or more authentication service objects 1310 to be included with the programs 12, 22 at all times, or the programs will not function. Alternatively, rules 140 may govern the acceptance of authentication service objects or object updates, for example requiring explicit approval from the user. Another approach is the use of a master authentication service object 1310 to authenticate all other authentication service objects 1310. This master authentication service object may be a built-in system object. It may also use a large public key or keys that are publicly verifiable via other trusted communications networks such as newspapers or telephones.

Detailed Description Text (400):

Any interested buyer can use the same classified ad service object 1310 and category object 110 to specify and monitor ad listings that meet the buyer's interests. The steps involved in this process are shown in FIG. 34B. (This process is similar to the process of monitoring category objects 110 on a directory partner server 1302 as shown in FIG. 31B.) As with the ad listing process, the monitoring process begins with the buyer using his/her browser 50 to navigate the classified ad partner server 1302. The buyer chooses the hyperlink representing the category object 110 in which the buyer is interested in making a purchase (step 4231). The receipt method for the category object 110 first checks to see if its parent classified ad service object 1310 is present in the consumer database 11 (step 4232). If not, the category object uses its link component object 110 to download the classified ad service object 1310 from the directory partner server 1302 (step 4233). The receipt method 141 then executes a data exchange method 141 in the classified ad service object 1310 that generates a monitoring input form (step 4234). The monitoring input form is largely identical to the listing input form described above. It draws some of its attributes and values from the category object 110. The principle difference is that it allows the buyer to specify value ranges or other query formulas for category attributes obtained from the category object 110. To use the automobile example above, a "Minivan" category object might use drop-down list of integer values for the "Not older than" year attribute; use checkboxes for multiple color choices; accept an integer value for "maximum mileage"; use radio buttons for acceptable condition attributes; and so on. When the form is submitted, the data exchange method 141 first saves the input form data as a query element 143 (step 4235). Secondly it creates one or more scheduled event instances 117 in the consumer database 11 (step 4236). These scheduled event instances 117 can begin immediately and repeat at intervals or according to rules 140 specified by the consumer on the input form. They can also be subject to monitoring rules 140 imposed by the classified ad service provider in the category object 110 or classified ad service object 1310. When activated, these scheduled event instances execute a data exchange method 141 in the classified ad service object 1310 (step 4237). The data exchange method 141 then creates a message object 110 containing the ad query (step 4238). The data exchange method 141 transmits this message object 110 to the classified ad partner server 1302 (step 4239). When received by the classified ad partner server 1302, the message object 110 triggers a corresponding data exchange method 141 (step 4240). This data exchange method 141 uses the ad query to query the classified ad partner server database 1301 for any ad listings satisfying the query (step 4241). The data exchange method 141 then returns the result set to the consumer program 12 (step 4242). If there are no ad listings that satisfy the query, the result set is a message object 110 reporting this. If there are ad listings that satisfy the query, the result set are the communications objects 110 representing the seller together with the component object 110 representing the seller's ad listing. This is advantageous because these communications objects 110 enable the buyer and seller to immediately establish

their own communications relationship to consummate the sale. After the consumer program 22 receives the result set, it executes any receipt methods pertaining to the result set objects (step 4243). This includes the notification test (step 4244). If there were no matching ads, the buyer may not desire notification. If there are matching ads, the buyer may desire different notification based upon the attributes values of the matching ads. For example, if a minivan meeting the buyer's query was listed at a price below a certain value, the buyer might desire to be paged immediately, whereas at a higher price the buyer may only wish to receive notification in the buyer's notification report (630, FIG. 13). The provider program 12 then executes any desired notification methods (step 4245). Again, this process could also incorporate authentication, payment, reporting, or other service object services.

Detailed Description Text (405):

A payment service object type (842, FIG. 17) is a specialized data exchange service object that operates in conjunction with payment partner servers 1302 to provide secure financial transaction services to providers and consumers. A payment service object 1310 may combine the functions of a data exchange service object 1310 with those of an authentication service object 1310, or it may call the services of a separate authentication service object 1310. (The examples in this section will use the latter technique.) Payment service objects allow such common payment services as credit card transactions, debit card transactions, electronic funds transfers, and cybercash transactions to take place easily, automatically, and securely in a communications object system.

Detailed Description Text (406):

The following explains the basic processes involved with the use of payment service objects 1310 and payment partner servers 1302. These are broken into several sets as shown in FIGS. 37, 38, and 39. The steps in the process of a merchant creating a payment account are illustrated in FIG. 37. The process begins with the merchant obtaining a copy of the payment service object 1310 if one is not already present in the provider database 11 (step 4401). When the merchant is ready to use the payment service object 1310 to set up a payment account, the merchant activates a data exchange method 141 in the payment service object 1310 (step 4402). This data exchange method 141 first generates a public/private key pair, either itself or by calling the services of an authentication service object 1310 (step 4403). Alternatively the data exchange method 141 can use an existing public/private key pair available from the authentication service object 1310. The private key is stored as an element 143 of the payment service object 1310 in the provider database 11 (step 4404). As with an authentication private key, this key may also be encrypted with a password known only to the user and not stored locally. The data exchange method 141 then queries the provider database 11 for the elements 143 necessary to create a payment account (step 4405). This process is explained in the data exchange control section. Because many of these elements 143 are commonly required items of data, such as the provider's name, contact data, financial account data, credit references, and so on, they will already be present in the provider database 11 and can be automatically accessed by the payment service object 1310. The data exchange method 141 then generates an account data input form (step 4406). The purpose of this form, as with most data exchange input forms, is threefold. First, it allows the merchant to confirm any data exchange rules 140 the merchant may have applied to the transfer of the merchant's sensitive financial data, such as bank account numbers or credit references. Second, it allows the merchant to confirm the accuracy of any other data to be transferred, such as contact data. Third, it allows the merchant to enter any specific new data required by the payment service provider. As explained in the data exchange control section, such new data can also be saved as elements in the provider database 11 for future use. When the merchant submits the completed input form, the data exchange method 141 creates a account order (step 4407). The payment service object 1310 then calls an authentication method 141 in an authentication service object 1310 to encrypt the account order using the payment partner server's public key, stored in the

payment service object 1310 as an element 143 (step 4408). The authentication method 141 then digitally signs the account order using the merchant's private key (step 4409). The merchant's private key may be stored in the authentication service object 1310 as an encrypted element 143, in which case the authentication method 141 may first require a password from the merchant for decryption. Alternatively the merchant's private key may be entered manually in some other way. The data exchange method 141 now creates a message object 110 containing the secure account order and the merchant's public key certificate, stored as an element 143 in the authentication service object 1310 (step 4410). Optionally this message object 110 may also contain such data as is necessary to create a user object 110 representing the merchant at the payment partner server 1302. The data exchange method 141 transmits this message object 110 to the payment partner server 1302 (step 4411). The payment partner server 1302 receives the message object 110 and executes a data exchange receipt method 141 (step 4412). This data exchange method 141 calls the same authentication service object 1310 to decrypt the secure account order using the payment partner server's private key, stored as an element 143 in the partner server database 1301 (step 4413). Next the authentication service object 1310 verifies the merchant's public key certificate signature using the authentication partner server's public key, stored in the authentication service object 1310 as an element 143 (step 4414). Finally the authentication service object 1310 verifies the merchant's signature on the account order using the merchant's public key (step 4415). Now the data exchange method 141 on the payment partner server 1302 can execute whatever steps are necessary to use the account order to create a merchant account (step 4416). In a preferred embodiment, the merchant account would be represented by a user object 110 in the payment partner server database 1301. When finished, the data exchange method 141 creates a merchant account certificate consisting of the merchant's account number, the merchant's provider UID, and whatever other data the payment service provider wishes to include in the certificate, such as a timestamp, account type identifiers, payment partner server identifiers, and so on (step 4417). (The merchant account certificate may also be encrypted if desired using a single key; decryption will only be necessary at the payment partner server.) The data exchange method 141 then calls an authentication method 141 in the authentication service object 1310 to digitally sign the merchant account certificate using the payment partner server's private key (step 4418). Next the data exchange method 141 creates a message object 110 containing the signed merchant account certificate (step 4419). The data exchange method 141 transmits this message object 110 back to the payment service object 1310 in the merchant's provider program 12 (step 4420). There the provider program 12 receives the message object 110 and executes the original data exchange method 141 of the payment service object 1310 (step 4421). This data exchange method 141 first calls an authentication method 141 in the authentication service object 1310 to verify the signature of the merchant account certificate using the payment partner server's public key (step 4422). Then the data exchange method 141 stores the merchant account certificate in the provider database 11 as an element 143 of the payment service object 1310 (step 4423). Lastly the data exchange method 141 calls any notification methods desired by the merchant for notification of the merchant account certificate receipt (step 4424). This completes the process of setting up a secure payment account for the merchant.

#### Detailed Description Text (407):

To begin using this account with customers, the merchant includes the merchant account certificate and a link component object 110 from the payment service object 1310 in any communications object 110 where the merchant wishes to use payment services. The payment service object 1310 can then be called by any data exchange method 141 in the merchant's communications object 110. The merchant can indicate the services of such payment service objects 1310 by using the names or logos of the appropriate credit cards, debit cards, and so on in a product ordering input form, for example. When a customer chooses one of these options and submits a data exchange input form, the payment service object 1310 is used automatically. The steps in this process are shown in FIG. 38. First the data exchange method 141

creates a purchase order consisting of the data from the input form together with the merchant account certificate (step 4441). Next the data exchange method 141 queries to see if the payment service object 1310 is present in the customer's consumer database 21 (step 4442). If not, the data exchange method 141 uses the payment service object's link component object 110 to download the payment service object 1310 (step 4443). The payment service object's receipt method 141 will then initiate the process to create a customer account (step 4444). This process is identical to the merchant payment account creation process shown in FIG. 37, except the final result is that the customer is issued a customer account certificate stored in the consumer database 21 as an element 143 of the payment service object 1310. If the payment service object 1310 was present in the consumer database 21 in step 4441, the data exchange method 141 calls a version monitoring method 141 to see if the version is current (step 4445). This version monitoring method 141 compares the version value of the payment service object 1310 with the version value stored in the link component object 110 of the merchant's communications object 110. Version monitoring is explained in the data exchange control section above. If the version is not current, the data exchange method 141 executes the update method 141 of the payment service object 1310 to download the current version (step 4456). Once the current version of the payment service object 1310 is present in the consumer database 21, the data exchange method 141 in the merchant's communications object 110 calls a data exchange method 141 in the payment service object 1310 to continue the transaction (step 4457). This data exchange method 141 calls an authentication method 141 in an authentication service object 1310 to encrypt the purchase order using the payment partner server's public key, stored in the payment service object as an element 143 (step 4458). The authentication method 141 also digitally signs the purchase order using the customer account certificate private key (step 4459). As described above, this key may be stored as an encrypted element 141 in the payment service object 1310 and require a password from the customer to decrypt. Alternatively the customer may supply the key manually in some other way. Next the data exchange method 141 creates a message object 110 containing the secure purchase order and the customer account certificate (step 4460). The data exchange method 141 transmits this message object 110 to the payment partner server 1302 (step 4461). The payment partner server 1302 receives the message object 110 and executes its receipt method 141, which is either the same data exchange method 141 or another data exchange method 141 residing on the payment partner server 1302 (step 4462). This data exchange method 141 calls an authentication method 141 in the authentication service object 1310 to verify the customer's signature on the secure purchase order using the customer account certificate public key (step 4463). Next the authentication method 141 decrypts the purchase order using the payment partner server's private key (step 4464). Finally the authentication method 141 verifies the merchant's signature on the merchant account certificate using the payment partner server's private key (step 4465). Now a data exchange method 141 on the payment partner server 1302 can carry out the purchase order transaction using the verified purchase order data, the verified customer account certificate, and the verified merchant account certificate (step 4466). This may involve any sequence of steps between the payment partner server 1302 and other payment servers or data processing systems, such as the consumer's bank or credit clearinghouse, a credit card processor, a cybercash server, and so on. When the transaction has been completed, the data exchange method 141 creates a unique receipt number stored as an element 143 in the payment partner server database 1301 (step 4467). This receipt number can now be used to verify the transaction with both the customer and the merchant.

Detailed Description Text (408):

From this point the receipt acknowledgment process can take several paths. The payment partner server 1302 can return receipt acknowledgments to both the consumer program 22 and the provider program 12. Each of these programs can in turn send receipt acknowledgments to the other to complete full three-way acknowledgment. Alternatively the payment partner server 1302 can send a receipt acknowledgment to the customer's consumer program 22, which can in turn send a receipt acknowledgment

to the merchant's provider program 12, or vice versa. In all cases the steps in sending secure receipt acknowledgment messages are similar. The steps in the process of the payment partner server 1302 sending a receipt acknowledgment message to the customer's consumer program 22 are shown in FIG. 39. First a data exchange method 141 on the payment partner server 1302 creates a purchase receipt (step 4471). The purchase receipt includes the unique receipt number plus any other relevant data, such as timestamp, the payment partner server UID, bank certification numbers, and so on. Next the data exchange method 141 calls an authentication method 141 in an authentication service object 1310 to encrypt the purchase receipt using the customer account certificate public key (step 4472). This step is optional if the purchase receipt does not contain any sensitive information. The authentication method 141 then signs the purchase receipt with the payment partner server's private key (step 4473). The data exchange method 141 creates a message object 110 containing the purchase receipt (step 4474). The data exchange method 141 transmits this message object 110 to the payment service object 1310 at the consumer program 22 (step 4475). There the consumer program 12 receives the message object 110 and executes the original data exchange method 141 of the payment service object 1310 (step 4476). This data exchange method 141 first calls an authentication method 141 in the authentication service object 1310 to verify the signature on the purchase receipt using the payment partner server's public key (step 4477). If the purchase receipt has been encrypted, the authentication method 141 decrypts it using the customer account certificate private key (step 4478). Then the data exchange method 141 stores the purchase receipt in the consumer database 21 as an element 143 of the payment service object 1310 (step 4479). This makes the purchase receipt available to the payment service object 1310 and the merchant communications object 110 for use in any further transactions or correspondence involving this transaction, such as a return or exchange. Finally the data exchange method 141 executes any notification methods 141 desired by the customer for notification about the receipt acknowledgment (step 4480).

Detailed Description Text (409):

This technique can be generalized to any form of data exchange requiring secure, verifiable, non-repudiable transactions between multiple parties. This includes stock trading, electronic data interchange (EDI), credit card systems, banking systems, bartering systems, and so on. The specific nature of the transaction service is not a limiting feature of the invention.

Detailed Description Text (414):

Anonymous reporting relationships can be accomplished using a simple procedure. The steps in this process are shown in FIG. 40. The process begins when a reporting service object 1310 first needs to set up an anonymous reporting relationship with a reporting partner server 1302. A data exchange method 141 in the reporting service object 1310 uses an anonymous protocol such as HTTP to request an anonymous reporting key from a reporting partner server 1302 (step 4501). The reporting partner server 1302 returns a unique anonymous reporting key in the protocol response (step 4502). The data exchange method 141 then saves the anonymous reporting key as an element 143 of the reporting service object 1310 (step 4503). If desired, such an element 143 can also be encrypted using a password or similar key provided by the user. From this point on the reporting service object 1310 can supply the anonymous reporting key together with the report data when submitting reports via the anonymous protocol to the reporting partner server 1302 (step 4504). In this fashion the reporting partner server 1302 can track the report data submitted from a unique instance of the reporting service object 1310 without having any knowledge of the user's identity (step 4505).

Detailed Description Text (423):

The value of feedback data can vary enormously with the experience and expertise of the feedback provider. This is particularly true for feedback on topics requiring specialized knowledge or expertise, such as academics, law, medicine, technology, and so on. For this reason feedback services can also be applied to feedback

providers. This can be accomplished using a feedback partner server 1302 by linking feedback category objects 110 to user objects 110 representing each of the feedback providers. The attributes of a feedback category object 110 representing a feedback provider might include level of expertise, level of credibility, level of decision-making ability, and so on. By aggregating feedback data on feedback providers, a feedback partner server 1302 is able to offer even more useful feedback reports to feedback consumers. This is because feedback queries can select feedback data using on the attributes or "ratings" of the feedback providers. An example is a feedback partner server 1302 which collects feedback data on communications objects 110 representing automobiles. A feedback consumer can create a query for only those communications objects 110 representing minivans with a sticker price of less than \$20,000 which also had overall quality rating of 7 or higher on a scale of 1 to 10 from feedback providers whose expertise level was rated by other feedback providers to also 7 or higher on a scale of 1 to 10. Another example applies to response thread objects (FIG. 29B) in a topic discussion database. Here a feedback consumer can use a topic feedback category object 110 to monitor the response thread objects 110 contained by a discussion topic 110. A query can notify the feedback consumer only of new response thread objects posted by providers with an expertise rating of 7 or higher on a scale of 1 to 10. A feedback consumer can also ask for feedback provider ratings to be factored into feedback data reports. An example would be a report on recommended minivans where the feedback data from feedback providers with an expertise rating of 8 or higher on a scale of 1 to 10 was weighted twice as heavily as feedback data from feedback providers with a rating lower than 8.

Other Reference Publication (60):

Budi Yuwono and Dik Lun Lee, "Wise: A World Wide Web Resource Database System", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. Aug. 1996.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#) [Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

☐ [Generate Collection](#) [Print](#)

L9: Entry 168 of 188

File: USPT

Jan 29, 2002

US-PAT-NO: 6343275

DOCUMENT-IDENTIFIER: US 6343275 B1

TITLE: Integrated business-to-business web commerce and business automation system

DATE-ISSUED: January 29, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wong; Charles	Los Altos Hills	CA	94022	

APPL-NO: 09/356327 [\[PALM\]](#)

DATE FILED: July 16, 1999

PARENT-CASE:

This application is a continuation, of application Ser. No. 08/995,591, filed Dec. 22, 1997 now U.S. Pat. No. 6,115,690.

INT-CL-ISSUED: [07] [G06 F 17/60](#)

US-CL-ISSUED: 705/26; 705/1, 705/7, 705/8, 705/9, 707/1, 707/10, 707/100, 707/102, 707/523, 707/217, 709/201

US-CL-CURRENT: [705/26](#); [705/1](#), [705/7](#), [705/8](#), [705/9](#), [707/1](#), [707/10](#), [707/100](#), [707/102](#), [709/201](#), [715/523](#)

FIELD-OF-CLASSIFICATION-SEARCH: 705/1, 705/7, 705/8, 705/9, 705/26, 707/1, 707/10, 707/100, 707/102, 707/217, 707/523, 709/201

See application file for complete search history.

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

[Search Selected](#)

[Search ALL](#)

[Clear](#)

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<a href="#">4882675</a>	November 1989	Nichtberger et al.	705/14
<input type="checkbox"/>	<a href="#">5237497</a>	August 1993	Sitarski	705/8
<input type="checkbox"/>	<a href="#">5311438</a>	May 1994	Sellers et al.	700/96
<input type="checkbox"/>	<a href="#">5353218</a>	October 1994	De Lapa et al.	705/14
<input type="checkbox"/>	<a href="#">5913061</a>	June 1999	Gupta et al.	709/310
<input type="checkbox"/>	<a href="#">5968110</a>	October 1999	Westrope et al.	705/27

☐ 5991739      November 1999      Cupps et al.      705/26

## FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	CLASS
996273	October 1999	EP	

## OTHER PUBLICATIONS

Business to Business on the Internet: Using the web to cut costs and build sales, Computer Reseller news pp 34, Nov. 1996.\*  
dialog reference file 9 00960974, Eric Clemons, Segmentation, differentiation, and flexible pricing: Experience with information technology and segment-tailored strategies, Journal of Management Information Systems: JMIS PP 9-36, 1994.\*  
dialog reference file 9 00960974, Eric Clemons, Segmentation, differentiation, and flexible pricing: Experience with information technology and segment-tailored strategies, Journal of Management Information Systems: JMIS PP 9-36.

ART-UNIT: 2162

PRIMARY-EXAMINER: Stamber; Eric W.

ASSISTANT-EXAMINER: Alvarez; Raquel

## ABSTRACT:

A software system business-to-business Web commerce (Web business, or e-business) and automates to the greatest degree possible, in a unified and synergistic fashion and using best proven business practices, the various aspects of running a successful and profitable business. Web business and business automation are both greatly facilitated using a computing model based on a single integrated database management system (DBMS) that is either Web-enabled or provided with a Web front-end. The Web provides a window into a "seamless" end-to-end internal business process. The effect of such integration on the business cycle is profound, allowing the sale of virtually anything in a transactional context (goods, services, insurance, subscriptions, etc.) to be drastically streamlined.

19 Claims, 339 Drawing figures

[Previous Doc](#)      [Next Doc](#)      [Go to Doc#](#)



[First Hit](#) [Fwd Refs](#) [Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

☐ [Generate Collection](#) [Print](#)

L9: Entry 172 of 188

File: USPT

Sep 5, 2000

US-PAT-NO: 6115690

DOCUMENT-IDENTIFIER: US 6115690 A

TITLE: Integrated business-to-business Web commerce and business automation system

DATE-ISSUED: September 5, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wong; Charles	Los Altos Hills	CA	94022	

APPL-NO: 08/995591 [\[PALM\]](#)

DATE FILED: December 22, 1997

INT-CL-ISSUED: [07] [G06 F 17/60](#)

US-CL-ISSUED: 705/7; 705/1, 705/8, 705/30, 705/34, 364/709.06, 364/479.07

US-CL-CURRENT: [705/7](#); [700/237](#), [705/1](#), [705/30](#), [705/34](#), [705/8](#), [708/136](#)

FIELD-OF-CLASSIFICATION-SEARCH: 235/380, 364/468.02, 364/468.14, 364/468.21, 364/479.06, 364/479.07, 364/479.08, 364/705.06, 364/709.06, 705/34, 705/1, 705/30, 705/7, 705/8

See application file for complete search history.

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

[Search Selected](#) [Search ALL](#) [Clear](#)

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<a href="#">5101352</a>	March 1992	Rembert	705/8
<input type="checkbox"/>	<a href="#">5191522</a>	March 1993	Bosco et al.	705/4
<input type="checkbox"/>	<a href="#">5224034</a>	June 1993	Katz et al.	705/7
<input type="checkbox"/>	<a href="#">5311438</a>	May 1994	Sellers et al.	364/468.02
<input type="checkbox"/>	<a href="#">5450317</a>	September 1995	Lu et al.	705/10
<input type="checkbox"/>	<a href="#">5528490</a>	June 1996	Hill	395/712
<input type="checkbox"/>	<a href="#">5557515</a>	September 1996	Abbruzzese et al.	705/9
<input type="checkbox"/>	<a href="#">5592378</a>	January 1997	Cameron et al.	705/27
<input type="checkbox"/>	<a href="#">5596502</a>	January 1997	Koski et al.	364/468.01
<input type="checkbox"/>	<a href="#">5615109</a>	March 1997	Eder	705/8

<input type="checkbox"/> <a href="#">5621201</a>	April 1997	Langhans et al.	235/380
<input type="checkbox"/> <a href="#">5638519</a>	June 1997	Haluska	705/28
<input type="checkbox"/> <a href="#">5666493</a>	September 1997	Wojcik et al.	705/26

ART-UNIT: 271

PRIMARY-EXAMINER: Cosimano; Edward R.

ASSISTANT-EXAMINER: Alvarez; Raquel

ATTY-AGENT-FIRM: Burns, Doane, Swecker & Mathis, LLP

ABSTRACT:

A software system business-to-business Web commerce (Web business, or e-business) and automates to the greatest degree possible, in a unified and synergistic fashion and using best proven business practices, the various aspects of running a successful and profitable business. Web business and business automation are both greatly facilitated using a computing model based on a single integrated database management system (DBMS) that is either Web-enabled or provided with a Web front-end. The Web provides a window into a "seamless" end-to-end internal business process. The effect of such integration on the business cycle is profound, allowing the sale of virtually anything in a transactional context (goods, services, insurance, subscriptions, etc.) to be drastically streamlined.

85 Claims, 129 Drawing figures

[Previous Doc](#)      [Next Doc](#)      [Go to Doc#](#)

[First Hit](#)   [Fwd Refs](#)   [Previous Doc](#)   [Next Doc](#)   [Go to Doc#](#)



Generate Collection

L9: Entry 173 of 188

File: USPT

May 30, 2000

US-PAT-NO: 6070160

DOCUMENT-IDENTIFIER: US 6070160 A

TITLE: Non-linear database set searching apparatus and method

DATE-ISSUED: May 30, 2000

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Geary; Wade S.	Salt Lake City	UT		

## ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Artnet Worldwide Corporation	New York	NY			02

APPL-NO: 08/593487   [\[PALM\]](#)

DATE FILED: January 29, 1996

## PARENT-CASE:

This application is a continuation-in-part of application Ser. No. 08/446,202, abandoned on Apr. 1, 1996.

INT-CL-ISSUED: [07] [G06](#) [F 17/30](#)US-CL-ISSUED: [707/4](#); [707/3](#), [707/5](#), [705/1](#), [382/149](#)US-CL-CURRENT: [707/4](#); [382/149](#), [705/1](#), [707/3](#), [707/5](#)FIELD-OF-CLASSIFICATION-SEARCH: [395/603](#), [395/604](#), [395/605](#), [395/606](#), [707/3](#), [707/5](#), [382/149](#), [705/1](#)

See application file for complete search history.

## PRIOR-ART-DISCLOSED:

## U.S. PATENT DOCUMENTS

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<a href="#">4999806</a>	March 1991	Chernow et al.	364/900
<input type="checkbox"/>	<a href="#">5099426</a>	March 1992	Carlgren et al.	395/759
<input type="checkbox"/>	<a href="#">5241671</a>	August 1993	Reed et al.	707/1
<input type="checkbox"/>	<a href="#">5263126</a>	November 1993	Chang	395/51
<input type="checkbox"/>	<a href="#">5379366</a>	January 1995	Noyes	395/54

<input type="checkbox"/>	<u>5379420</u>	January 1995	Ullner	707/6
<input type="checkbox"/>	<u>5384894</u>	January 1995	Vassiliadis et al.	395/61
<input type="checkbox"/>	<u>5544256</u>	August 1996	Brecher et al.	382/149
<input type="checkbox"/>	<u>5634051</u>	May 1997	Thomson	395/605
<input type="checkbox"/>	<u>5884272</u>	March 1999	Walker et al.	705/1

ART-UNIT: 271

PRIMARY-EXAMINER: Black; Thomas G.

ASSISTANT-EXAMINER: Rones; Charles L.

ATTY-AGENT-FIRM: Brown Raysman Millstein Felder & Steiner LLP

ABSTRACT:

A method and apparatus for creating, storing, identifying, transferring, managing and searching databases of information related to subjective works, such as art, music, film, dance, theater or other fields generally recognized as requiring subjective human judgment by an "expert" to make subjective, objective, or mixed decisions regarding value, interest, and relationships one to another. The invention may be embodied in a general purpose digital computer programmed to host routines operating by deterministic logic, fuzzy logic, or both. A user may input information related to the nature or type of item requested and receive identification of a subjective match for the item. The routines may utilize a thesaurus and processes for relaxing search requirements to assure a match. In one embodiment, an expert system resident in a computer may create, manage and rapidly search databases of subjectively characterized items, such as art works, music, or real estate, for example, by unique characteristics. The invention may include a method for securely transmitting data corresponding to a result, to prevent storage by a recipient computer. For example, an image may be represented in a self-executing, self-destructive, standard 7 bit ASCII text E-mailable packet without attachments.

33 Claims, 33 Drawing figures

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#)   [Fwd Refs](#)   [Previous Doc](#)   [Next Doc](#)   [Go to Doc#](#)

☐ [Generate Collection](#)   [Print](#)

L9: Entry 175 of 188

File: USPT

Oct 19, 1999

DOCUMENT-IDENTIFIER: US 5970482 A

TITLE: System for data mining using neuroagents

Drawing Description Text (10):

FIGS. 9(a)-(d), through various screen displays, show the process for creating a Subject in an embodiment of the present invention.

Drawing Description Text (11):

FIGS. 10(a)-(d), through various screen displays, show the process for creating a Study in an embodiment of the present invention.

Drawing Description Text (12):

FIG. 11 shows a screen display of the Scenario Specification.

Drawing Description Text (13):

FIG. 12 shows a screen display of the Population Description.

Drawing Description Text (14):

FIGS. 13(a)-(b) show screen displays of data distributions.

Drawing Description Text (15):

FIG. 14 shows a screen display of the Relationship Discovery Report.

Drawing Description Text (16):

FIG. 15 shows a screen display of the Conjunction Profile View.

Drawing Description Text (17):

FIG. 16 shows a screen display of the Specific/Irrelevant Criteria Profile View.

Drawing Description Text (18):

FIG. 17 shows a screen display of the Decision Impacts View.

Drawing Description Text (19):

FIG. 18 shows a screen display of the Evaluation Report.

Drawing Description Text (20):

FIG. 19 shows a screen display of the Top "N" Profile View.

Drawing Description Text (21):

FIG. 20 shows a screen display of the Study Improvement Report.

Drawing Description Text (22):

FIG. 21 shows a screen display of the Scoring Results.

Drawing Description Text (23):

FIG. 22 shows a screen display of the Mailing Results.

Drawing Description Text (24):

FIGS. 23(a)-(b) show screen displays of the Interactive Simulation Dialog and Simulation Influence Dialog, respectively.

Detailed Description Text (11):

Available through the DataMind menu 2066 but not otherwise shown is the dictionary, which contains equivalents between the generic data mining language and a user's own language. The dictionary may also be used to establish data equivalence, criterion equivalence or qualitative values, as assigned to Discovery results. An example of data equivalence, in the setting of a sales analysis, could be that code "1" represents the Western Region (such association already having been made in the Data Warehouse environment, Data Warehousing to be discussed below). An example of criterion equivalence could be that the first category of a parameter, age (1:age), has a synonym which is "teenager." The synonym is the descriptor assigned to an input or output. This descriptor tells more about the output in relation to other outputs. For example, if a set of outputs describe the "best" to "worst" outcomes for an objective, one may describe them as "very good," "good," "average," "bad," and "very bad" using synonyms. The various results screens will display this descriptive label, as appropriate, based on the results. Also, discovery levels are by default quantitative values that can be changed in a custom and qualitative way. For example, one could define a "very low" importance level to be between the values 0 and 20. Illustrations of these features will be found in the description that follows.

Detailed Description Text (14):

Generally, a Data Warehouse acts as a central data store for operational data extracted and transformed from other data storage platforms, such as Online Transaction Processing (OLTP) systems. A Data Warehouse is a relational database management system (RDBMS) designed specifically for decision-making rather than transaction processing. Another way of defining a Data Warehouse is that it is a superset of historical data collected from different operational data bases and processed for uniformity of storage and retrieval for business purposes. Although OLTP systems can also be an RDBMS, they are optimized for day-to-day operations rather than ad hoc information retrieval and business analysis. Furthermore, their data models are normalized, i.e., the database schema are written with a minimum of redundancy, such as having only one key field to identify a particular record in a table. Data Warehouses maintain multi-dimensionality to retain versatility in their ad hoc queries. While OLTP's contemplate a changing and incomplete data landscape, Data Warehouses are created for a historical and descriptive purpose. One example is Red Brick.TM. Warehouse VPT, produced by Red Brick Systems of Los Gatos, Calif., which uses RISOQL.TM., a Data Warehouse-adapted extension to the SQL relational database language originally developed by IBM Corp. of Armonk, N.Y. The Red Brick.TM. Warehouse VPT System is adapted for very large data warehouses in excess of 500 Megabytes, parallel query processing on symmetric multiprocessing architectures, and time-based data management. RISOQL.TM. adds Data Warehouse functionality to SQL such as cumulative totals, moving averages and sums, ranking results rows including breaking up into tertiles, as well as evaluating ratios of values to totals. Clearly, this is advantageously suited to provide updated information which can test the predictions of a Data Mining Tool.

Detailed Description Text (20):

It should be understood that multiple Data Sources may be aggregated into a Subject in of two ways: (i) the records of each Data Source merely accumulated with the others and so arranged consecutively; or (ii) the records of one or more Data Sources merged into larger records. In the latter case, care must taken to properly align the data records from the multiple Data Sources, much as an inner join is performed between tables in a single relational database. The Subject is thus the coherent assembly of Data Sources, Parameter for Parameter.

Detailed Description Text (62):

However, for other variables, such as "VISA.sub.-- CREDI" (VISA.TM. credit card rating), the distribution may be more suitably displayed as a pie chart 2275 (or histogram or other such familiar graphical construct), as in FIG. 13(b). Thus, with

the VISA.sub.-- CREDI distribution may be readily seen as "very low" 2280, "low" 2281, "middle level" 2282, "high" 2283 and "very high" 2284, with an additional legend 2285 useful for making the association.

Detailed Description Text (82):

One should note that a "criterion" is a singular concept that the user can manipulate directly in a decision-making process. For example, "PRICE" alone is not what one would normally manipulate to make a decision; rather, one might use "PRICE LOW" and "PRICE HIGH" instead. In this case, "PRICE" is the field, "LOW" is one possible value for the field "PRICE" and "PRICE LOW" is the criterion. It is not necessarily the same as the data presented in the database, instead being regrouped into a more meaningful set of data. As example: with data which refers to children aged between 1 and 3 years, "toddler" could be the criterion.

Detailed Description Text (89):

Importance, as shown in column 2210, expresses the relative impact of a criterion on a decision, i.e., the impact of the input, or the impact of the conjunction to which the input belongs, against the output. Built during the Discovery process, the importance regroups all the Discovery relations such as the specificity (i.e., the more specific the relation is, the higher the impact), the discrimination (i.e., the more discriminating the relation is, the higher the impact is), the noise threshold, the conjunction(s), etc.

Detailed Description Text (90):

Thus, in this example, the importance of the criteria, "Occupation Principal," "Merchant level 8," and "Merchant level 7," is maximal, at 100%, for the output, "ACCOUNT Balanced." In this embodiment, the maximal importance of a criterion is 100% times the number of criteria in the conjunction in which this particular criterion forms part. Thus, the criteria cited in this example are not in conjunction.

Detailed Description Text (135):

With reference now to FIG. 19, the Top "N" Profile View 2600 is shown. Thus, for the output ACCOUNT, there are results for "Balanced" 2610, "Overdraft" 2620 and "VISA Late Payment" 2630, with "+" buttons 2615 and 2625 for showing more than the default three profiles for output results "Balanced" 2610 and "Overdraft" 2620, respectively. Thus, the columns represent the variables for this example Study, namely ACCOUNT 2640 (output), presented with the line number of the record 2642, SEX 2643, MARITALS 2644 (marital status), CHILDREN 2645 (number of children), OCCUPATION 2646, HOME 2648, EXPENSES 2650, INCOME 2652, CHECKING 2654, SAVINGS 2656, MSTRCARD 2658 (MasterCard.TM. rating), VISA.sub.-- CREDIT 2659 (VisaCard.TM. rating), AMEX 2660 (American Express.RTM. rating), MERCHANT 2662 (merchant rating), PAYMENTTHIS 2664 (payment history, in months), with the Excitation Level 2666 and Discrimination 2668. Thus, for example, one can see that the data entry that registered most strongly as ACCOUNT "Visa Late Payment", i.e., had an Excitation Level of 621, has an OCCUPATION of "Professional" and a VISA.sub.-- CREDIT of "middle level."

Detailed Description Text (161):

Thus, in the example given here, the results are shown with the now familiar fields SEX 2902, MARITALS 2904 (marital status), CHILDREN 2906 (number of children), OCCUPATION 2908, HOME 2910, EXPENSES 2912, INCOME 2914, CHECKING 2916, SAVINGS 2920, MSTRCARD 2922 (MasterCard.TM. rating), VISA.sub.-- CREDIT 2924 (VisaCard.TM. rating), AMEX 2926 (American Express.TM. rating), MERCHANT 2928 (merchant rating), PAYMENTTHIS 2930 (payment history, months), along with the predicted output ACCOUNT 2932 and the scoring level 2934.

Detailed Description Text (164):

Thus, in the example given here, the columns appear identically as in the Scoring Results, with fields SEX 3302, MARITALS 3304, CHILDREN 3306, OCCUPATION 3308, HOME

3310, EXPENSES 3312, INCOME 3314, CHECKING 3316, SAVINGS 3318, MSTRCARD 3320, VISA.sub.-- CREDIT 3322, AMEX 3324, MERCHANT 3326 and PAYMENTTHIS 3328. With the Mailing Results 3300, however, the prediction of the output ACCOUNT 3330 is specified by the user, here as "Balanced", and the rows are thus ordered by the scoring level 3332.

Detailed Description Text (182):

The user interface 3080 allows the user to interact with the application 3000 and may include a display screen, keyboard, mouse, printer, or other such peripherals. The spreadsheet compilation unit 3084 prepares most tabular data created by the system 3000 for display. The graphics compilation unit 3088 prepares most graphical data created by the system 3000 for display.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)



[First Hit](#)   [Fwd Refs](#)   [Previous Doc](#)   [Next Doc](#)   [Go to Doc#](#)



Generate Collection

L9: Entry 175 of 188

File: USPT

Oct 19, 1999

US-PAT-NO: 5970482

DOCUMENT-IDENTIFIER: US 5970482 A

TITLE: System for data mining using neuroagents

DATE-ISSUED: October 19, 1999

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pham; Khai Minh	Menlo Park	CA		
Rajkovic; Eric Bertrand	Foster City	CA		
Piffero; Veronique	Menlo Park	CA		

## ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Datamind Corporation	San Mateo	CA			02

APPL-NO: 08/600229   [\[PALM\]](#)

DATE FILED: February 12, 1996

INT-CL-ISSUED: [06] [G06 F 15/18](#)

US-CL-ISSUED: 706/16; 706/12, 706/45

US-CL-CURRENT: [706/16](#); [706/12](#), [706/45](#)

FIELD-OF-CLASSIFICATION-SEARCH: 395/10, 395/22, 395/77, 395/50, 706/12, 706/45, 706/16

See application file for complete search history.

PRIOR-ART-DISCLOSED:

## U.S. PATENT DOCUMENTS

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<a href="#">5398300</a>	March 1995	Levey	706/16
<input type="checkbox"/>	<a href="#">5586218</a>	December 1996	Allen	706/12
<input type="checkbox"/>	<a href="#">5615341</a>	March 1997	Agrawal et al.	705/10
<input type="checkbox"/>	<a href="#">5692107</a>	November 1997	Simoudis et al.	706/12

## OTHER PUBLICATIONS

Intelligent Data Analysis Methods in DataEngine, <http://ss-m3.mitmgbh.de/mit/products/demos.html/dedemo.sub.--3.zip>, Sep. 1994.

IntelliSphere.COPYRG.T., "User's Guide & Reference Guide" (Version 2.0), 1995, pp. I-23 to I-53, II-51 to II-82, II-197 to II-270.

K. M. Pham, "The NeurOagent: A Neural Multi-agent Approach for Modelling, Distributed Processing and Learning," Intelligent Hybrid Systems, 1995, pp. 221-244.

R. Kerber, B. Livezey, E. Simoudis, "A Hybrid System for Data Mining," Intelligent Hybrid Systems, 1995, pp. 121-142.

S. Goonatilake, S. Khebbal, "A Hybrid Systems Classification Scheme," Intelligent Hybrid Systems, pp. 7-11., 1995.

A. Szladow and W. Ziarko, "Rough Sets: Working with Imperfect Data," AI Expert, pp. 36-41, Jul. 1993.

P.D. Varhol, "Modeling systems with polynomial networks: tools for predicting behavior," Dr. Dobb's Journal, vol. 18(9), p. 76(5), Sep. 1993.

D.S. Barr and G. Mani, "Using Neural Nets to Manage Investments," AI Expert, pp. 16-21, Feb. 1994.

W. Dwinnell, "The Second Generation Cometh: Advanced Modeling Systems," AI Expert, pp. 38-41, Jun. 1994.

Neural Networks Resource Guide, AI Expert, pp. 42-51, Jun. 1994.

"The Power of Database Mining," AI Expert, p. 46, Dec. 1994.

"Greater Data Insight," AI Expert, p. 41, Feb. 1995.

"Go Ahead and Make a Prediction," AI Expert, p. 45, Jun. 1995.

Database Mining Workstation, HNC, Inc., Software Product Specification, Dec. 1991.

"Mining, minus mining mishaps" (Product Showcase), AI Expert, p. 54, Aug. 1992.

C.J. Matheus, et al., "Systems for Knowledge Discovery in Databases," IEEE Trans. on Knowledge and Data Engineering, vol. 5(6), pp. 903-913, Dec. 1993.

DataLogic/R (V.1.3), REDUCT Systems, Inc., Software Product Specification, Dec. 1993.

L. Lewinson, "Data mining: tapping into the mother lode," Database Programming & Design, vol. 7(2), pp. 50(5), Feb. 1994.

J. Angstenberger, et al., "DataEngine: A Software Tool for Intelligent Data Analysis," WESCON/94, pp. 348-350, Sep. 1994.

DataLogic/R+ (V.1.5), Reduct Systems, Inc., Software Product Specification, Dec. 1994.

R. Sharpe, "A mine of information," Computer Weekly, pp. 33(2), Jan. 1995.

"Pushing the data mining envelope," AI Expert, p. 46, Jan. 1995.

M. Marshall, "New edition of data-mining tool makes up its own rules," Communications Week, n.559, p. 12(2), May 1995.

W. Pickering, "Cognos calls Angoss for data-mining power," PC Week, vol. 12(31), pp. 33(2), Aug. 1995.

P.K. Chan and S.J. Stolfo, "Learning Arbiter and Combiner Trees from Partitioned Data for Scaling Machine Learning," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 39-44, Aug. 1995.

A. Ciampi and Y. Lechevallier, "Designing Neural Networks from Statistical Models: A new approach to data exploration," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 45-50, Aug. 1995.

C. Cortes, et al., "Capacity and Complexity Control in Predicting the Spread Between Borrowing and Lending Interest Rates," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 51-56, Aug. 1995.

C. Cortes, et al., "Limits on Learning Machine Accuracy Imposed by Data Quality," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 57-62, Aug. 1995.

R. Kohavi and D. Sommerfield, "Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 192-197, Aug. 1995.

H.-Y. Lee, et al., "Exploiting Visualization in Knowledge Discovery," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 198-203, Aug. 1995.

E. Simoudis, et al., "Using Recon for Data Cleaning," Proc. First Int'l. Conf. on Knowledge Discovery & Data Mining, pp. 282-287, Aug. 1995.

N. Zhong and S. Ohsuga, "Toward A Multi-Strategy and Cooperative Discovery System," Proc. First int'l. Conf. on Knowledge Discovery & Data Mining, pp. 337-342, Aug. 1995.

"Mining data," PC Week, p. E5, Aug. 1995.

R. Shortland and R. Scarfe, "Digging for gold," IEE Review, vol. 45(5), pp. 213-217, Sep. 1995.

W.W. Eckerson, "Information Harvesting offers data mining and forecasting tool," Distributed Computing Monitor, vol. 10(9), p. S6, Sep. 1995.

I. Greenberg and C. Whitmer, "SPSS 7.0 for Windows 95 set to mine data warehouse systems," InfoWorld, vol. 17(42), p. 37, Oct. 1995.

A.J. Fridlund, "Sophisticated Statistica is slick jack-of-all-trades; wealth of features in easy to access," InfoWorld, vol. 17(44), p. 106, Oct. 1995.

S.R. Hedberg, "The Data Gold Rush," BYTE, pp. 83-88, Oct. 1995.

K. Watterson, "A Data Miner's Tools," BYTE, pp. 91-96, Oct. 1995.

C.D. Krivda, "Data-Mining Dynamite," BYTE, pp. 97-103, Oct. 1995.

B. Phillips, "Data-mining tool taps parallel servers," PC Week, pp. 36(2), Jan. 1996.

ART-UNIT: 272

PRIMARY-EXAMINER: Downs; Robert W.

ATTY-AGENT-FIRM: Wilson Sonsini Goodrich & Rosati

#### ABSTRACT:

A neuroagent approach is used in an automated and unified data mining system to provide an explicitly predictive knowledge model. The neuroagent is a neural multi-agent approach based on macro-connectionism and comprises a double integration at the association and symbolic level as well as the knowledge model level. This data mining system permits discovery, evaluation and prediction of the correlative factors of data, i.e., the conjunctions, as corresponding to neuroexpressions (a semantic connection of neuroagents) connected to an output neuroagent which corresponds to the data output, the connection weights yielding the relative significance of these factors to the given output. The system takes data sets called Domains, establishes candidate dimensions or Parameters, categorizes Parameters into discrete bins, and trains a neuroagent network composed of neuroagents allocated for each bin and each output based on a discovery data set, called a Discovery Domain, and by building up the various minimal and contextual neuroexpressions, and setting the appropriate connection weights, the results may therefore be compared with an optional evaluation data set, called an Evaluation Domain to establish the accuracy of the knowledge model, and thereafter applied with some degree of confidence to a prediction set or Prediction Domain. The ranking in importance of the composite Parameters may be calculated as well as the discrimination between the various outputs, which permits the relevant factors of interest to a decision maker to come into focus.

33 Claims, 41 Drawing figures

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set



Generate Collection

Print

L5: Entry 6 of 6

File: USPT

Jan 19, 1999

DOCUMENT-IDENTIFIER: US 5862325 A

TITLE: Computer-based communication system and method using metadata defining a control structure

Drawing Description Text (19):

FIG. 17 illustrates the object oriented database structures for different communications object types.

Detailed Description Text (18):

Information can be stored in the provider and consumer databases 11, 21, transferred between the provider and consumer programs 12, 22, and processed by these programs in a variety of ways. The use of software objects and object-oriented databases, and in particular their ability to encapsulate data and methods for operating on that data in a single structure, provide certain degrees of functionality which are useful in the storage, transfer, and processing of information. For example, by using objects for transmission of the communications control files, and an object-oriented database for storage of these files, the received object can be stored by the consumer program 22 in its database 21 without having to disconnect and store the object's variables and methods independently. In addition, the data and methods of this object can be made available to other objects in the database or program for processing operations. Object oriented data structures, databases, programs, and processing are generally discussed in Grady Booch, Object Oriented Analysis and Design with Applications, (2nd ed. 1994) and James Rumbaugh, Object-Oriented Modeling and Design (1991), which are incorporated herein by reference. Thus, the following description of a preferred embodiment will discuss the use of objects. However, other methods for storing, transferring, and processing information, such as relational databases, binary files, or procedural programs, could be used.

Detailed Description Text (59):

In order to transfer a communications object instance or object update instance from a provider program 12 to a consumer program 22, the object must be output from the provider database 11 into a format suitable for transport via a communications network 3. Any type of machine readable and writable format could be used, for example a compressed binary file such as that used by most relational or object-oriented database management programs. However, for maximum compatibility with communications networks 3 and other data processing systems, object instances can be written or read in an ASCII markup language, which is a superset of HTML. As with HTML, or other standard markup languages such as SGML, each item of structured data such as an object class or container class is expressed within a set of delimiters or "tags" defined in the markup language. Certain classes in the database structure exist specifically to provide the necessary container tags for other classes. For example, in FIG. 3, the methods 131, pages 132, elements 133, and type definitions 134 classes are all special container classes used to provide the tags necessary to delimit the methods, pages, elements, and type definition sections of an object output in the markup language. Another advantage of the use of an ASCII markup language is that the data and methods contained in communications object may be rendered readable to other data processing programs

for purposes of interoperability. Other programs may also be programmed to output such a language or a subset thereof for purposes of importing into a communications object system program. The use of an ASCII markup language does not preclude the use of additional formatting or encoding, such as encryption, for the entire object or for portions of the object.

Detailed Description Text (105):

One advantage of the communications system of the present invention is that the transmitted communications object instance can be automatically received, processed, stored, and indexed by the consumer program 22. Since the data is structured as an object and stored in an object-oriented database 21, the data it contains can be easily searched using the consumer program 22 in order to locate specific information or perform certain functions.

Detailed Description Text (193):

One particular advantage of a communications object system in this respect is the ease with which multiple public keys may be used. Multiple keys may be included within a single communications object, or a single key may be constantly changed via communications object updates, or both techniques can be used together. Since encryption can be applied automatically by the consumer program 22, the encryption method 141 can programmatically or randomly chose from among the available public keys. By including an identifier value 161 within each public key element 143, and including this unencrypted identifier value in the header of the encrypted message objects 110, the provider program 12 can also automatically identify and apply the matching private key element 143 for decryption. The use of multiple rotating public keys significantly reduces the risk of security breaches if any one key combination is broken, and increases the effort necessary to compromise the security of the messages.

Detailed Description Text (253):

As with any multiuser database system, shared access to data requires data access controls. This control should cover all common data operations such as creating, reading, writing, moving, and deleting data. In a communications object system, data access controls need to extend beyond human operators to communications objects, since these objects are essentially acting as "surrogates" for their respective providers. The key data structure involved with data access control is the rules class 140. Data access rules can monitor all forms of data access within the provider database 11 or consumer database 21 as well as external data in the provider or consumer's computing or network environment. For example, a typical rule governing access to communications object components or element preference instances might be that only other communications objects sharing the same database system ID (100, FIG. 3) can read, write, or delete such instances. This would prevent different providers from having access to each other's private data. This rule could be modified so that only communications objects sharing a group system ID (251, FIG. 6A), described above, could have access to such data. This would allow all communications objects created by employees of the same company, or within a company division, to access each other's communications object component or element preference instances. Data access rules can be system-wide, assigned by providers, or assigned by consumers. An example of a provider-assigned rule would be restrictions on communications object forwarding, which will be further discussed below. An example of a consumer-assigned rule would be that designated personal data, such as household income, must be explicitly authorized by the consumer before it is transmitted in any data exchange. A stricter rule would state that more sensitive private data, such as credit card numbers, must be encrypted and require one or more passkeys for decryption prior to any data exchange. In order to protect their integrity, data access rules can also enforce the ability to add or change other data access rules, and also the hierarchy in which rules take precedence when two or more rules apply. Data access rules can also be selectively applied by the consumer to particular communications objects 110 or communications object groups such as folders 115 by creating associations between these and a data

access rule 140. The application of rules to control data access within an active database is further discussed in the aforementioned Active Database Systems.

Detailed Description Text (384):

Many cryptographic protocols have been devised to provide authentication of user identity and message integrity over data networks. These include Kerberos 5, developed at MIT; SPX, developed by Digital Equipment Corporation; Privacy Enhanced Mail (PEM), adopted by the Internet Engineering Task Force (IETF); Pretty Good Privacy (PGP), developed by Philip Zimmermann; and the CCITT X.509 protocols. Such protocols are fully described in the aforementioned Applied Cryptography by Bruce Schneier. Authentication service objects 1310 and authentication partner servers 1302 can be employed to automate the operation of many of these protocols. This is accomplished by storing the appropriate encryption keys as elements 143 and the appropriate encryption functions as methods 141 of the authentication service object 1310 or authentication partner server 1302.

Detailed Description Text (385):

An example is authentication using digital signatures based on public/private keys. The first set of steps in this process are shown in FIG. 32A. The process begins with the provider obtaining a suitable authentication service object (1310, FIG. 28) if one is not already present in the provider program 12 (step 4101). An authentication service object 1310 contains one or more public keys from its corresponding authentication partner server 1302, stored as elements 143. The authentication service object 1310 also contains the encoding method or methods 141 necessary to carry out its authentication functions, called authentication methods. When the provider is ready to create an authentication account, the provider executes one of the authentication methods 141 to generate a public/private key pair (step 4102). The private key is stored as an element 143 of the authentication service object 1310 in the provider database 11 (step 4103). Optionally, the data exchange method 141 may also encrypt this private key element 143 with a password known only to the provider and not stored anywhere in the provider database 11 or on the local computer. The authentication method 141 next creates an authentication order consisting of three elements: the public key generated in step 4102, the provider's UID, and a unique registration key known only to the provider and the authentication partner server 1302 (step 4104). Other elements or variables, such as a timestamp, may also be included. If the authentication partner server 1302 is operated in conjunction with a registration partner server 1302, the unique registration key may be the provider's password or other identification key created at the time of registration. This is shown as the Key attribute of the system ID instance (251, FIG. 6A). This unique registration key is stored in the provider database 11 as an encrypted element 143 which can be decrypted using a provider-supplied password. Alternatively, it may not be stored at all locally but be entered manually by the provider when required. The authentication method 141 next encrypts the authentication order using the authentication partner server's public key (step 4105). The authentication method 141 then creates a message object 110 containing the encrypted authentication order (step 4106). The authentication method 141 transmits this message object 110 to the authentication partner server 1302 (step 4107). The authentication partner server 1302 receives the message object 110 and executes its receipt method 141, which is either the same authentication method or another authentication method residing on the authentication partner server 1302 (step 4108). This authentication method 141 decrypts the authentication order using the authentication partner server's private key (step 4109). Next the authentication method 141 verifies the provider's unique registration key and UID in the authentication partner server database 1301 to validate the authentication order (step 4110). The authentication method 141 then creates a public key certificate by combining the provider's public key with certain other identifying data, such as the provider's UID (step 4111). The authentication method 141 digitally signs the public key certificate using the authentication partner server's private key (step 4112). The authentication method 141 then creates a message object 110 containing the public key certificate (step 4113). Finally, the

authentication method 141 transmits the message object 110 back to the authentication service object 1310 in the provider program 12 (step 4114). There the provider program 12 receives the message object 110 and executes the original authentication method 141 in the authentication service object 1310 (step 4115). This authentication method 141 first verifies the signature of the public key certificate using the public key of the authentication partner server 1302 (step 4116). Lastly the authentication method 141 saves the public key certificate in the provider database 11 as an element 143 (step 4117).

Detailed Description Text (389):

Authentication on a communications object system may also take place without using centralized authentication partner servers 1302. This technique, known as distributed key management, is used by the public-domain encryption program Pretty Good Privacy (PGP). It is based on the concept of an "introducer". An introducer is a person who signs the public key certificate of another person whose identity they personally know and are willing to certify. Introducers are easily employed on a communications object system using authentication service objects 1310. The steps in the process for using introducers are illustrated in FIG. 33A. First, a user requiring a public key certificate introduction, called the "originator", executes a data exchange method 141 of an authentication service object 1310 to generate a public/private key pair (step 4151). Next, the data exchange method 141 stores each key as an element 143 of the authentication service object 1310 (step 4152). Then the data exchange method 141 creates a public key certificate consisting of the public key element 143 plus such additional elements 143 as will allow any potential introducer to certify the identify of the orginator (step 4153). These first three steps can be omitted if the originator only wishes to add introducers for an existing public key certificate already stored as an element 143 of the authentication service object 1310. Now, the data exchange method 141 generates an input form prompting the originator for the recipients 120 whom the originator would like to make introduction requests (step 4154). The checkboxes on this input form can represent each of the recipients 120 in the originator's consumer database 21, or the originator can specify the e-mail addresses of still other potential introducers. The input form also allows the originator to enter the attributes of a message element (211, FIG. 4) to be sent to these recipients. When the input form is submitted, the data exchange method 141 creates a message object 110 consisting of the public key certificate, the message element, and any other relevant data, such as a timestamp (step 4155). The data exchange method 141 transmits this to all recipients 120 selected by the originator (step 4156). When received by the recipient's consumer program 22, the message object's receipt method 141 executes the recipient's selected notification method or methods 141 for introduction requests (step 4157). If distributed key management was implemented on a communications object system, message objects containing introduction requests can use a standard notification element type definition 144. This type definition 144 allows consumers to assign notification methods 141 globally for all introduction requests, or designate specific notification methods for introduction requests from individual recipients 120. When the recipient responds to the notification message, a data exchange method 141 in the authentication service object 1310 is executed (step 4158). This data exchange method 141 generates a input form for confirming the introduction request from the originator (step 4159). This input form may include any such data as may be relevant to an introduction request, including the elements 143 of the public key certificate that fully identify the originator. The recipient may also wish to verify the public key with the originator via another secure channel, such as via telephone. When the recipient is satisified that the request is genuine, the recipient submits the input form (step 4160). The data exchange method 141 calls an authentication method 141 in the authentication service object 1310 which digitally signs the originator's public key certificate using the recipient's private key (step 4161). If the recipient's private key is stored as an encrypted element 143 of the authentication service object 1310, the recipient may need to enter password or passphrase for decryption. Then the data exchange method 141 creates a message object 110 containing the signed public key

certificate (step 4162). The data exchange method 141 transmits this message object 110 to the originating authentication service object 1310 at the originating consumer program 22 (step 4163). When the message object 110 is received, the consumer program 22 executes the originating data exchange method 141 (step 4164). This data exchange method 141 stores the signed public key certificate as an element 143 of the authentication service object 1310 (step 4165). Finally, the data exchange method 141 executes any notification methods 141 assigned by the originator to the acknowledgment of introduction requests (step 4166).

Detailed Description Text (405):

A payment service object type (842, FIG. 17) is a specialized data exchange service object that operates in conjunction with payment partner servers 1302 to provide secure financial transaction services to providers and consumers. A payment service object 1310 may combine the functions of a data exchange service object 1310 with those of an authentication service object 1310, or it may call the services of a separate authentication service object 1310. (The examples in this section will use the latter technique.) Payment service objects allow such common payment services as credit card transactions, debit card transactions, electronic funds transfers, and cybercash transactions to take place easily, automatically, and securely in a communications object system.

Detailed Description Text (407):

To begin using this account with customers, the merchant includes the merchant account certificate and a link component object 110 from the payment service object 1310 in any communications object 110 where the merchant wishes to use payment services. The payment service object 1310 can then be called by any data exchange method 141 in the merchant's communications object 110. The merchant can indicate the services of such payment service objects 1310 by using the names or logos of the appropriate credit cards, debit cards, and so on in a product ordering input form, for example. When a customer chooses one of these options and submits a data exchange input form, the payment service object 1310 is used automatically. The steps in this process are shown in FIG. 38. First the data exchange method 141 creates a purchase order consisting of the data from the input form together with the merchant account certificate (step 4441). Next the data exchange method 141 queries to see if the payment service object 1310 is present in the customer's consumer database 21 (step 4442). If not, the data exchange method 141 uses the payment service object's link component object 110 to download the payment service object 1310 (step 4443). The payment service object's receipt method 141 will then initiate the process to create a customer account (step 4444). This process is identical to the merchant payment account creation process shown in FIG. 37, except the final result is that the customer is issued a customer account certificate stored in the consumer database 21 as an element 143 of the payment service object 1310. If the payment service object 1310 was present in the consumer database 21 in step 4441, the data exchange method 141 calls a version monitoring method 141 to see if the version is current (step 4445). This version monitoring method 141 compares the version value of the payment service object 1310 with the version value stored in the link component object 110 of the merchant's communications object 110. Version monitoring is explained in the data exchange control section above. If the version is not current, the data exchange method 141 executes the update method 141 of the payment service object 1310 to download the current version (step 4456). Once the current version of the payment service object 1310 is present in the consumer database 21, the data exchange method 141 in the merchant's communications object 110 calls a data exchange method 141 in the payment service object 1310 to continue the transaction (step 4457). This data exchange method 141 calls an authentication method 141 in an authentication service object 1310 to encrypt the purchase order using the payment partner server's public key, stored in the payment service object as an element 143 (step 4458). The authentication method 141 also digitally signs the purchase order using the customer account certificate private key (step 4459). As described above, this key may be stored as an encrypted element 141 in the payment service object 1310 and require a password from the



customer to decrypt. Alternatively the customer may supply the key manually in some other way. Next the data exchange method 141 creates a message object 110 containing the secure purchase order and the customer account certificate (step 4460). The data exchange method 141 transmits this message object 110 to the payment partner server 1302 (step 4461). The payment partner server 1302 receives the message object 110 and executes its receipt method 141, which is either the same data exchange method 141 or another data exchange method 141 residing on the payment partner server 1302 (step 4462). This data exchange method 141 calls an authentication method 141 in the authentication service object 1310 to verify the customer's signature on the secure purchase order using the customer account certificate public key (step 4463). Next the authentication method 141 decrypts the purchase order using the payment partner server's private key (step 4464). Finally the authentication method 141 verifies the merchant's signature on the merchant account certificate using the payment partner server's private key (step 4465). Now a data exchange method 141 on the payment partner server 1302 can carry out the purchase order transaction using the verified purchase order data, the verified customer account certificate, and the verified merchant account certificate (step 4466). This may involve any sequence of steps between the payment partner server 1302 and other payment servers or data processing systems, such as the consumer's bank or credit clearinghouse, a credit card processor, a cybercash server, and so on. When the transaction has been completed, the data exchange method 141 creates a unique receipt number stored as an element 143 in the payment partner server database 1301 (step 4467). This receipt number can now be used to verify the transaction with both the customer and the merchant.

Detailed Description Text (409):

This technique can be generalized to any form of data exchange requiring secure, verifiable, non-repudiable transactions between multiple parties. This includes stock trading, electronic data interchange (EDI), credit card systems, banking systems, bartering systems, and so on. The specific nature of the transaction service is not a limiting feature of the invention.

Detailed Description Text (414):

Anonymous reporting relationships can be accomplished using a simple procedure. The steps in this process are shown in FIG. 40. The process begins when a reporting service object 1310 first needs to set up an anonymous reporting relationship with a reporting partner server 1302. A data exchange method 141 in the reporting service object 1310 uses an anonymous protocol such as HTTP to request an anonymous reporting key from a reporting partner server 1302 (step 4501). The reporting partner server 1302 returns a unique anonymous reporting key in the protocol response (step 4502). The data exchange method 141 then saves the anonymous reporting key as an element 143 of the reporting service object 1310 (step 4503). If desired, such an element 143 can also be encrypted using a password or similar key provided by the user. From this point on the reporting service object 1310 can supply the anonymous reporting key together with the report data when submitting reports via the anonymous protocol to the reporting partner server 1302 (step 4504). In this fashion the reporting partner server 1302 can track the report data submitted from a unique instance of the reporting service object 1310 without having any knowledge of the user's identity (step 4505).

Other Reference Publication (55):

K. Smith and S. Zdonik "Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems" OOPSLA '87 Proceedings.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)